



Métaphores de présentation de l'information basées sur des méthodes procédurales

Noémie Esnault, Jérôme Royan, Rémi Cozot, Christian Bouville

► To cite this version:

Noémie Esnault, Jérôme Royan, Rémi Cozot, Christian Bouville. Métaphores de présentation de l'information basées sur des méthodes procédurales. AFIG 2009, Nov 2009, Arles, France. inria-00479790

HAL Id: inria-00479790

<https://inria.hal.science/inria-00479790>

Submitted on 2 May 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Métaphores de présentation de l'information basées sur des méthodes procédurales

Noémie Esnault^{1,2}, Jérôme Royan², Rémi Cozot¹, et Christian Bouville¹

¹Université de Rennes 1, IRISA

²Orange Labs

Abstract

Aujourd'hui, le Web 3D se limite encore à des univers 3D accessibles depuis une application autonome indépendamment du contenu Web, et progressivement via des modules d'extension dédiés pour navigateurs Web. Des standards en cours de spécification comme HTML 5 ou WebGL permettront dans un avenir proche d'intégrer, voir de fusionner des contenus Web 2D avec des contenus 3D. Un défi majeur consistera alors à créer des mondes 3D permettant la visualisation et la navigation à l'intérieur de données multimédia sans nécessairement maîtriser les outils de modélisation complexes qui ne permettent pas de générer des environnements 3D pouvant évoluer automatiquement selon les modifications des données et/ou informations à afficher. Nous proposons ici un environnement logiciel permettant de décrire une métaphore de présentation de l'information, prenant en considération la structuration des données, c'est à dire comment l'utilisateur explore et découvre les informations, ainsi que l'aspect visuel, c'est à dire la géométrie du monde 3D et la représentation des données ou informations.

Nowadays, 3D on the Web is limited to 3D worlds that require a standalone software independent of the Web content, or uses Web browser's add-ons. New standards, such as HTML 5 or WebGL, are in a specification phase and will soon be available, allowing the integration on the Web of 2D content with 3D content. A major challenge will then be to be able to create 3D worlds for the visualization and browsing of multimedia content, without needing to master professional 3D authoring tools, which do not allow the computation of 3D worlds evolving automatically with the information or data to be visualized. In this article, we present a software environment for the definition of information's display metaphors, taking both the structuring of data (how the user explores and discover data) and the visual aspect of the computed environment (the geometry and data representation) into account.

1. Introduction

Le concept de Web3D, introduit dès 1994 par le consortium du même nom avec la spécification du VRML, était précurseur, mais malheureusement en avance de phase. Débits réseaux trop faibles, capacités graphiques limitées des terminaux, interfaces de navigation et d'interaction peu accessibles, l'émergence du Web3D n'a pas eu lieu. Cependant, ces dernières années, des mondes virtuels en ligne comme Second Life, Google Earth et bien d'autres ont démontré la faisabilité d'un déploiement industriel de ces technologies 3D en ligne. De plus, l'intérêt montré par les industriels concernant les interfaces intuitives d'interaction et de navigation (Natal project, OpenViBE, ...) ajouté à la volonté de la communauté Web de spécifier et développer des tech-

nologies 3D (WebGL, HTML 5, ...), laissent présager une émergence d'un Web3D alternant interfaces 3D de présentation de l'information et environnements virtuels immersifs. Naturellement dans la lignée du Web2.0, le Web3D sera probablement auto-produit, nécessitant des outils de modélisation de scènes virtuelles qui soient assistés et accessibles au plus grand nombre d'utilisateurs. Dans ce contexte, nous présentons dans cet article une solution de modélisation automatisée et personnalisable de scènes 3D permettant la visualisation d'une grande quantité d'information issue de bases de données ou de résultats de recherche classique ou sémantique.

Notre environnement logiciel permet de décrire des mé-

taphores de présentation de l'information prenant en considération :

- la structure exploratoire des données qui va conditionner la manière dont l'utilisateur va explorer et découvrir les informations
- l'aspect visuel, c'est à dire la représentation des données ou des informations sous forme d'un monde 3D.

La construction de la métaphore de structuration, appelée "Genotype computation", consiste à construire un graphe acyclique orienté décrivant la structuration exploratoire des données. La construction de la métaphore visuelle, appelée "Phenotype computation", organise le Génotype dans l'espace 3D à l'aide de grammaires procédurales basées sur des règles de réécriture.

Les principales caractéristiques de notre environnement sont :

- une approche basée XML avec l'utilisation de feuilles de styles permettant une séparation forte entre les données et leur visualisation
- la compatibilité avec les données issues de bases de données, avec l'exploitation des données sémantiques et/ou des moteurs de recherche classiques ou sémantiques
- la séparation entre les données et leur visualisation qui se caractérise par la possibilité de réutilisation des Phénotypes pour des données/contenus différents, et réciproquement des Génotypes avec un passage très facile d'une métaphore de visualisation à une autre pour la présentation de données.

La suite de cet article est organisée comme suit. Nous commencerons par une présentation de l'état de l'art couvrant d'une part les technologies Web telles que le XML, les feuilles de style et les langages de transformation du XML, et d'autre part les grammaires procédurales utilisées pour construire automatiquement des environnements 3D dédiés. La troisième partie décrit en détails notre environnement d'auteur en commençant par une vue globale, puis en détaillant le contenu et l'exploitation du Génotype et du Phénotype. La quatrième partie est dédiée aux résultats, nous montrons comment à partir de bases de données multimédia nous construisons différentes métaphores de présentation. Enfin, nous concluons en indiquant les perspectives d'évolution de notre environnement.

2. Etat de l'art

2.1. Technologies Web

Depuis son apparition, le Web a connu de nombreuses transformations. Au delà de l'évolution matérielle des réseaux permettant aux utilisateurs d'échanger toujours plus de données grâce aux débits de plus en plus élevés, la révolution du Web a nécessité une évolution et une diversification des outils de création et une spécialisation des concepts sur lesquels il se base. A l'origine du Web, aucun outil de création n'étant

disponible, la production de sites Web était réservée aux programmeurs. Avec l'arrivée d'outils d'aide à la création de sites Web, tout utilisateur, même le moins expérimenté, peut participer activement à la création de contenu sur le Web. La dissociation entre le contenu et sa présentation a grandement contribué à l'adoption massive du Web, devenu Web 2.0. De nouveaux outils de création sont apparus (pour la production de blogs par exemple), et la personnalisation des contenus (commentaires, notation, tags, portail Web personnalisable, ...) comme des aspects visuels (couleurs et thèmes des portails Web personnalisables, des webmails, ...) ont participé à l'appropriation du Web par les utilisateurs.

Aussi, l'interopérabilité des outils et concepts du Web, assurée par des standards tels que HTML (Hypertext Markup Language) ou XML (eXtensible Markup Language) ont fortement favorisé l'avènement du Web. Les chapitres suivants donnent un état de l'art des principales technologies Web utilisées pour permettre la séparation entre le contenu des sites Web et leur aspect visuel.

2.1.1. EXtensible Markup Language

De nombreux langages utilisés sur le Web respectent la syntaxe XML (EXtensible Markup Language). Cette recommandation du World Wide Web Consortium (W3C, [Web]) ([BPSM*98]) a été initialement conçue pour le stockage, le transport, le partage et la structuration de données. Il s'agit d'un métalangage pour la création de langages à balises. Les noms des éléments et attributs XML ne sont pas prédéfinis mais spécifiés pour chaque nouveau langage créé. Les langages obtenus sont extensibles (par exemple par l'ajout de nouveaux noms) sans perte de compatibilité. La structure et les contraintes des éléments et attributs d'un langage basé sur XML peuvent être définis à l'aide d'une DTD (Document Type Definition), ou d'XML Schema (alternative XML aux DTD). Un document XML sera alors *valide* par rapport à un schéma si les éléments et attributs du document XML satisfont les contraintes spécifiées par le schéma.

XML est un métalangage auto-descriptif permettant de se concentrer sur ce que sont les données. Il s'agit d'un fichier texte Unicode, ce qui lui permet d'être très léger (et donc facilement partageable sur un réseau), mais aussi indépendant aussi bien d'un point de vue matériel que logiciel. C'est donc un format privilégié pour stocker et décrire des informations, mais aussi pour partager des données entre applications.

Sur le Web, de nombreuses technologies telles que XHTML, WAP, RSS ou RDF respectent la syntaxe XML. Les avantages d'interopérabilité et d'extensibilité d'XML lui ont également permis de se développer en dehors du Web (OpenDocument, OpenXML, XML Query, UIML, ...).

2.1.2. Aspects visuels et feuilles de style

Si XML permet de décrire et de partager des données, il ne permet pas de définir un mode de présentation de

l'information qu'il contient. La technologie la plus fréquemment utilisée pour définir le style graphique d'une page Web dont le contenu est décrit en XML correspond aux feuilles de style en cascade ou CSS (Cascading Style Sheets). Ce standard publié par le W3C permet de décrire l'aspect visuel en dehors du document contenant les données. Les styles ainsi décrits sont appliqués au document au moment de l'affichage, grâce à une transformation effectuée par l'application utilisée (un navigateur Web par exemple). Les dernières spécifications de CSS incluent une fonctionnalité de transformation 3D (CSS 3D Transform), permettant de transformer dans un espace 3D les éléments rendus par CSS. Cependant, aucune spécification ne permet d'insérer de modèles 3D paramétriques ou non comme éléments visuels.

2.1.3. Langages de transformation

Les langages de transformation XML permettent de créer toute forme de documents (XML, texte, binaire, ...) à partir d'un ou de plusieurs fichiers XML. Le plus connu, XSLT (eXtensible Stylesheet Language Transformations, [Kay07]), est une recommandation du W3C s'appuyant sur XPath pour désigner la partie d'un arbre XML à traiter. Le document XML en entrée n'est pas modifié lors de la transformation. XSLT est fréquemment utilisé pour convertir des données XML en HTML en effectuant par exemple une transformation dynamique sur le client ou le serveur. XSLT est également utilisé pour traduire des documents XML entre différents schémas XML.

2.1.4. Formats 3D du Web

Qu'ils soient issus des outils de modélisation 3D (.3ds, .ma, ...), des moteurs de rendu (.mesh, .osg, ...), des plates-formes propriétaires de jeux ou de mondes virtuels (Second Life, World of Warcraft), de formats d'échange (.fbx, ...) ou de consortiums de visualisation (.vrm, .dae, .x3d, .mp4, ...), la multiplicité des formats 3D n'a malheureusement pas contribué à l'émergence du concept de Web3D. Afin d'améliorer la compatibilité avec les formats Web et de permettre son extensibilité, le format X3D (eXtensible 3D, [Con04]) basé sur XML succéda au VRML. D'autres solutions poussées par la communauté Web, comme PaperVision3D ou Swift3D basé sur Flash, ainsi que de nouveaux formats en cours de spécification comme WebGL ou HTML 5.0 présagent l'adoption massive de représentations 3D sur le Web, que ce soit comme interface de présentation de l'information, ou sous forme de mondes virtuels immersifs.

2.2. Grammaires procédurales

De nombreuses études ont été menées pour présenter divers informations provenant de bases de données, sous forme d'environnements 3D navigables. [SC07] proposent une méthode de visualisation d'un cursus universitaire, où les différents éléments (départements, modules, ...) sont

représentés par des blocs dont la taille et la couleur sont fonction d'informations issues d'une base de données (durée du module, crédits, ...). De même, [WCW06] proposent une méthode permettant de créer des expositions dans des musées virtuels. Le nombre de salles, leurs caractéristiques et les objets qu'elles présentent sont également issus d'une base de données, et la méthode proposée permet de personnaliser la visualisation obtenue (définitions de nouvelles salles, de nouveaux décors, ...).

Les solutions proposées sont toujours extrêmement liées aux données à afficher et ne sont pas aisément adaptables pour visualiser d'autres types de contenu. En effet, le format de la base de données est fixé et aucune structuration générique du résultat des requêtes n'est proposé.

3. Vue d'ensemble

3.1. Objectifs

Nous souhaitons proposer un formalisme permettant de définir une métaphore complète de visualisation de données hiérarchiques. Ainsi, à partir d'un ensemble quelconque de données, nous proposons d'extraire le résultat d'une requête sous forme de hiérarchie. Chaque niveau de cette hiérarchie, appelé par la suite *niveau d'exploration*, permet d'explorer les données obtenues avec une granularité différente. Si nous prenons l'exemple d'une base de données de films, un utilisateur pour spécifier dans sa requête l'ordre des niveaux d'explorations (par exemple le genre, suivi du réalisateur, puis le titre du film, voir exemple 1). Une métaphore personnalisable est ensuite appliquée au résultat de la requête afin de visualiser les données sous forme d'un ensemble d'environnements 3D. L'ensemble d'environnements a les caractéristiques suivantes:

- sa *topologie* correspond à la structure hiérarchique du résultat de la requête
- sa *géométrie* correspond à la métaphore visuelle choisie

L'exemple suivant, qui servira de référence dans la suite de l'article, présente un cas d'utilisation de notre système.

Exemple 1 Nous souhaitons visualiser les résultats d'une requête effectuée sur une base de données multimédia contenant des films et les informations associées (réalisateur, genres, acteurs, durée, date de sortie, affiche, ...). L'utilisateur précisera comment il souhaite organiser les films de la base de données, par exemple selon les critères suivants: Genre, puis Réalisateur, puis Titre du film. Pour visualiser la réponse à cette requête, nous choisissons la métaphore *Tour de carrousels* composée de trois niveaux (correspondant ici aux 3 critères choisis): les éléments obtenus par le premier critère (les genres) seront représentés par des cylindres blancs et un texte correspondant au nom de l'élément. Les cylindres seront répartis linéairement selon l'axe Y de manière à former une tour. Les éléments obtenus par le deuxième critères (ici, réalisateurs) seront représentés par des sphères réparties circulairement autour du cylindre

correspondant. Enfin, les éléments du troisième critère (les titres de films) seront représentés par un cadre 3D dont la partie centrale représente l'affiche du film. Le résultat que nous souhaitons obtenir est présenté en figure 1.

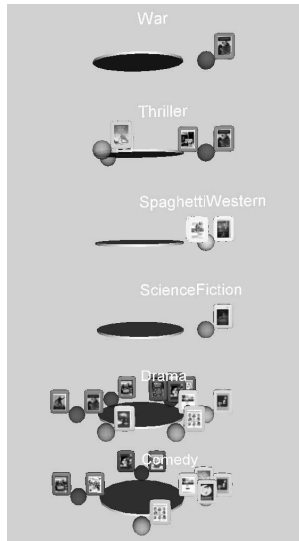


Figure 1: Exemple 1: tour de carrousels représentant une base de données de films.

Afin de visualiser les données dans leur ensemble, il est nécessaire que la métaphore visuelle soit compatible avec la structure hiérarchique choisie. Par exemple, le nombre de niveaux d'exploration pris en compte par la métaphore visuelle doit être supérieur ou égal au nombre de niveaux d'explorations de la structure hiérarchique obtenue. Le choix et la personnalisation de la structure hiérarchique et de la métaphore permettent d'obtenir une visualisation correspondant aux besoins de l'utilisateur, au contexte, et adaptable aux préférences utilisateurs.

3.2. Architecture

Afin d'atteindre les objectifs fixés en section 3.1, nous proposons un formalisme basé sur deux composants principaux, permettant ainsi de séparer la métaphore de structuration de la métaphore visuelle. Ces deux composants sont appelés respectivement Genotype Computation pour la structuration des données, et Phenotype Computation pour leur visualisation (voir figure 2).

L'architecture proposée peut être intégrée dans un outil exploitant toute la puissance de ce formalisme tout en préservant la simplicité de sa mise en œuvre. En effet, chaque règle définie dans les métaphores peut être considérée comme un module. Chaque module peut ensuite être imbriqué avec d'autres modules pour former une nouvelle métaphore. Ainsi, un utilisateur expert pourra définir de nouveaux modules, un utilisateur intermédiaire pourra combiner plusieurs

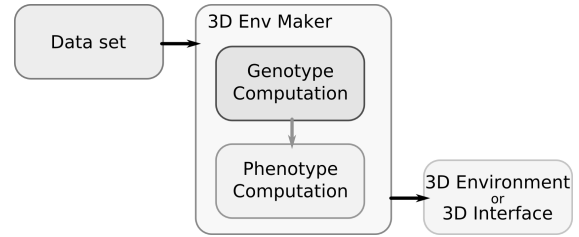


Figure 2: Architecture globale.

modules ou métaphores (comme une tour, un carrousel) pour créer une nouvelle métaphore (une tour de carrousels, un carrousel de tours, ...), tandis qu'un utilisateur occasionnel pourra choisir sa métaphore dans un catalogue, et la personnaliser à l'aide des paramètres proposés (couleur, taille, ...).

Le Genotype correspond à une représentation hiérarchique et pertinente des données. Il est généré à partir d'une requête multi-critères sur l'ensemble de données proposé en entrée du système. Chaque élément du Génotype peut être enrichi par des labels et des métadonnées (objets 3D, textures, couleurs, nombres, ...) permettant de le représenter au moment de la visualisation. Ainsi, dans l'exemple 1, les réalisateurs sont représentés par une couleur tandis que les films sont représentés par leur affiche. Le squelette du Génotype (ordre des niveaux d'exploration) peut être défini par l'utilisateur afin de mieux correspondre à ses attentes. Il peut par exemple choisir l'ordre des critères (genre, réalisateur, titre ou genre, date de sortie, titre), ou l'ordonnancement des données (ordre alphabétique, préférences utilisateur, ...).

Le Phenotype correspond à la définition du style géométrique qui sera appliqué au Génotype. Il est composé de règles de distributions d'une part, et de paramètres correspondant aux éléments distribués d'autre part. Dans l'exemple 1, les règles de distribution sont les suivantes:

- niveau d'exploration 1: distribution linéaire (tour)
- niveau d'exploration 2: distribution circulaire (carrousels de niveau 1)
- niveau d'exploration 3: distribution circulaire (carrousels de niveau 2)

Les niveaux d'exploration correspondent aux niveaux hiérarchiques du Génotype. Ils permettent de visualiser le résultat de la requête à différents niveaux d'abstraction. Chaque niveau d'exploration peut être visualisé de manière indépendante (un environnement 3D par niveau d'exploration, comment dans l'exemple présenté en section 6.3). On obtient alors une information de plus en plus précise au fur et à mesure de la visualisation des environnements. Il est également possible de visualiser plusieurs niveaux d'exploration dans un unique environnement (voir exemple 1). Le résultat obtenu grâce à notre système est

donc un ensemble d'environnements 3D navigables permettant la visualisation personnalisée de l'ensemble des données issues de la requête, associées à des informations spécifiques à chaque environnement (des arbres dans une ville, ...). La topologie de chaque environnement correspond à un sous-ensemble de la hiérarchie du Génotype, tandis que sa géométrie correspond aux critères définis dans le Phénotype. Une fois la métaphore de présentation définie, elle peut être appliquée à n'importe quel type de requête sur un ensemble de données et les métaphores visuelles et de structuration peuvent être combinées de différentes manières.

L'architecture que nous présentons sépare organisation des données et métaphore de visualisation : différentes métaphores de visualisation peuvent être appliquées à la même requête, et une métaphore peut être utilisée pour visualiser diverses requêtes provenant d'ensembles de données différents. Dans les chapitres suivants, nous décrivons plus en détails les structures du Génotype et du Phénotype.

4. Génotype

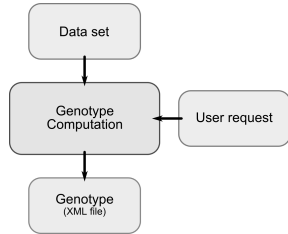


Figure 3: Architecture du Génotype.

Le Génotype (voir figure 3) correspond à une représentation sous forme de fichier XML du résultat d'une requête multi-critères sur un ensemble de données. Ce dernier peut être sémantique (utilisation de RDF ou des cartes topiques par exemple), ou non. L'utilisateur peut définir deux types de critères de recherches : des critères hiérarchiques permettant d'organiser les données et des critères de métadonnées permettant d'enrichir le contenu du Génotype généré. Chaque critère hiérarchique (mot clé EXPLO) correspond à un niveau d'exploration des données et définit la topologie de l'environnement. Les critères de métadonnées sont des attributs (mot clé ATT) pouvant être associés à un niveau d'exploration ou à un élément à afficher. Ils correspondent aux données multimédias (image, maillage 3D, ...) ou leurs attributs (couleurs, ...) pouvant être utilisés par le Phénotype pour représenter l'élément auquel ils sont attachés. Les attributs sont typés et peuvent contenir divers valeurs (url, ...). Des *labels* (mot clé TAG) peuvent également être associés aux attributs pour ajouter des informations sémantiques ou autre à l'attribut.

Définition 1 Un *Genotype* est une grammaire générative telle que :

- $V = \{GENOTYPE, EXPLO, ELEM, ATT, TAG\}$, tel que
 $GENOTYPE$ correspond à la racine de la hiérarchie,
 $EXPLO$ correspond à un niveau d'exploration,
 $ELEM$ correspond à un élément résultat de la requête sur l'ensemble de données,
 ATT est un attribut pouvant être utilisé par le Phénotype,
 TAG est un label (sémantique ou autre).
- $\omega = GENOTYPE$.
- P possédant les règles de construction suivantes :
 $GENOTYPE \rightarrow EXPLO + ATT^* + TAG^*$
 $EXPLO \rightarrow ELEM^{+*} + ATT^* + TAG^*$
 $ELEM \rightarrow ATT^* + TAG^* + EXPLO^{0|1}$
 $ATT \rightarrow TAG^*$.

Chaque symbole peut avoir divers paramètres, comme par exemple leur identifiant (id) interne à l'ensemble de données. Les attributs (ATT) possèdent toujours un type (couleur, entier, url, ...) pouvant être accompagné d'autres paramètres, comme les composants rouge vert et bleu pour une couleur.

L'exemple 2 présente un extrait d'un Génotype généré à partir d'une base de données de films, avec comme critères hiérarchiques le genre, puis les réalisateurs et enfin les titres. Le résultat de la requête est enrichi à l'aide des critères de métadonnées suivants: une couleur par réalisateur, et l'url d'une image représentant l'affiche du film pour chaque titre.

Exemple 2

```

< GENOTYPE name = "VOD" >
  < EXPLO name = "Genre" >
    < ELEM name = "Comedy" >
      < EXPLO name = "Directed by" >
        < ELEM name = "WoodyAllen" >
          < ATT type = "color" r = "0" g = "1" b = "1" >
            < TAG > background < /TAG >
          < /ATT >
        < EXPLO name = "Movie" >
          < ELEM name = "Another Woman" >
            < ATT type = "2Dimg" url = ". /pics / eightWomen. jpg" >
              < TAG > poster < /TAG >
            < /ATT >
          < /ELEM >
          < ELEM name = "Hollywood Ending" > ... < /ELEM >
        < /EXPLO >
      ...
    < /EXPLO > < /ELEM >
    < ELEM name = "Drama" > ... < /ELEM > < /EXPLO >
  < /GENOTYPE >
    
```

5. Phénotype

Le module Genotype Computation permet à l'utilisateur de sélectionner et d'ordonner ses données suivant différents critères. Une fois le génotype généré, il reste à définir la façon dont les données vont être affichées à l'écran. Le module Phenotype Computation (voir figure 4) permet la description d'un style géométrique qui sera appliqué au Génotype, permettant la création d'un ensemble d'environnements 3D. A l'aide de la grammaire formelle que nous proposons, l'utilisateur peut définir un grand nombre d'éléments constituant l'aspect visuel de l'environnement, comme le type et la

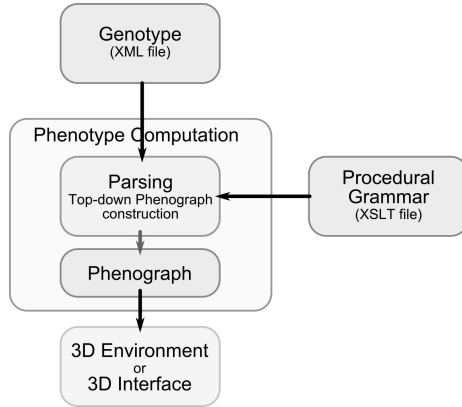


Figure 4: Phenotype architecture.

taille du contenu permettant de représenter chaque élément du Génotype (cube, sphère, texte, maillage 3D, texture, film, ...), la façon dont les éléments sont distribués dans l'espace (distribution circulaire ou linéaire, ...), ainsi que le contenu spécifique à l'environnement (couleurs, modèles 3D, terrain, ...) et permettant d'enrichir la visualisation obtenue et de mieux immerger l'utilisateur dans la métaphore choisie. L'analyse syntaxique parallèle du Phénotype et du Génotype permet de calculer à la volée un ensemble de graphes de scènes que nous appelons Phénographes. Les sections suivantes présentent plus en détails le Phénographe, puis le Phénotype et ses attributs.

5.1. Phénographe

Le Phénographe est le graphe de scène résultat de la combinaison d'un Phénotype et d'un Génotype. Il peut être implémenté dans différents formats (VRML, Collada, ...) et peut contenir des informations spécifiques permettant de lier ses objets à la base de données. Dans les exemples présentés dans ce document, nous avons choisi le format X3D pour le Phénographe. Ce standard permet entre autre la création de prototypes paramétrables. Les cadres (voir figure 5) utilisés dans l'exemple 1 sont des prototypes paramétrables dont la couleur et la texture sont spécifiées dynamiquement en fonction de métadonnées présentes dans le Génotype.

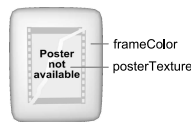


Figure 5: Prototype X3D et ses paramètres.

5.2. Phénotype

Le Phénotype permet de décrire le style géométrique qui sera appliqué aux données obtenues lors du calcul du Génotype.

Il est défini à l'aide d'un ensemble de règles de réécriture du Génotype composé de 5 catégories principales d'opérateurs :

- Structures de contrôle conditionnelles et itératives (boucles, conditions, ...)
- Interrogation multi-niveaux du Génotype (récupération des métadonnées correspondant au niveau d'exploration à visualiser, ou des niveaux d'explorations inférieurs et supérieurs)
- Insertion de modèles multimédia paramétriques ou non (images, vidéos, texte, maillages 3D, ...)
- Transformations 3D (translation, rotation, mise à l'échelles, ...)
- Découpage du résultat en un ensemble lié de fichiers de sortie (ajout dynamiques de liens, ...)

L'exemple 3 décrit la règle permettant d'afficher le deuxième niveau d'exploration du Génotype généré dans le cadre de l'exemple 1 : des sphères colorées représentant les réalisateurs pour chaque genre. Les sphères sont réparties circulairement autour du cylindre représentant le genre.

Exemple 3

```

Carrousel(nivDEExploration)
{
  pour tout elem ∈ nivDEExploration
  {
    rotation de ((360°/nbElem(nivDEExploration))*indice(elem)) autour de l'axe Y
    translation de 15 le long de l'axe X
    Insérer sphère(rayon=2, couleur=(métadonnée de elem de type 'color')
    si ∃ prochainNivDEExploration ∈ elem
    {
      Insérer règleSuivante(prochainNivDEExploration)
    }
  }
}
    
```

5.3. Implémentation

Nous avons choisi d'utiliser XSLT pour l'implémentation du module Phenotype Computation. En effet, il s'agit d'une technologie largement utilisée par la communauté Web pour la transformation de fichiers XML. De plus, ce langage possède nativement certains opérateurs nécessaires à la définition d'un Phénotype : boucles de parcours des données du fichier d'entrée (Génotype) et spécification des données sélectionnées, structures de contrôle, découpage possible du résultat en un ensemble de fichiers de sortie. Nous définissons alors trois types principaux de templates :

- Template de niveau d'exploration : permet de définir le décor général du niveau d'exploration et son contenu spécifique
- Template d'élément : permet de définir le contenu spécifique de l'élément
- Template nommé : cette fonctionnalité XSLT est utilisée pour la définition de modules. Ses paramètres peuvent être initialisés par l'intermédiaires d'autres templates.

Les templates nommés permettent d'obtenir une forte réutilisabilité des Phénotypes en facilitant l'imbrication de plusieurs Phénotypes afin de générer un nouveau Phénotype. Ainsi, si nous possédons déjà un template permettant d'effectuer une distribution linéaire, nous pouvons créer

un Phénotype de distribution bilinéaire en imbriquant deux Phénotypes de distribution linéaire.

Les modèles multimédia affichés peuvent être définis par le Phénotype (comme les cylindres blancs de l'exemple 1), par les métadonnées du Génotype (comme le nom des genres dans l'exemple 1), ou définis en partie par le Phénotype, avec une initialisation de paramètres à l'aide de métadonnées du Génotype (comme les sphères de l'exemple 1 permettant de représenter les réalisateurs dont le modèle 3D est défini dans le Phénotype, et la couleur est définie à l'aide du Génotype). L'utilisation d'XPath pour l'analyse syntaxique du Génotype permet de récupérer des métadonnées appartenant à des éléments autres que l'élément actuel (comme dans l'exemple 1 où la couleur du cadre d'un film est définie par le réalisateur, et donc l'élément du niveau d'exploration supérieur). Dans les exemples présentés, nous avons choisi le format de sortie X3D pour la création des Phénographes. Les opérateurs de transformation dans l'espace 3D et d'insertion de modèles multimédia sont donc implémentés à l'aide des spécifications X3D correspondantes. L'exemple 4 donne le résultat XSLT de la règle définie dans l'exemple 3.

Exemple 4 Extrait du Phénotype XSLT correspondant au niveau d'exploration 2 de la tour de carrousels (exemple 1) : distribution circulaire de sphères.

```
< xsl : stylesheet version = "1.0" >
...
< xsl : template match = /GENOTYPE/EXPLO/ELEM/EXPLO >
  < xsl : variablename = "nExplo" select = "@name" / >
  < xsl : variable name = "n" select = "count(./ELEM)" / >
  < Group def = "$nExplo" >
    < xsl : for - each select = "ELEM" >
      < xsl : variable name = "r" select = "(position() - 1) * 2 * $pi div $n" / >
      < xsl : variablename = "nElem" select = "@name" / >
      < xsl : variable name = "c" select = ".//ATT[@type = 'color']" / >
      < Transform rotation = "0 1 0 $r" >
        < Transform translation = "15 0 0" rotation = "0 1 0 - $r" >
          < Group def = "$nElem" >
            < Shape >
              < Sphere radius = "2" / >
              < Appearance >
                < Material diffuseColor = "$c/@r$c/@g$c/@b"
                  transparency = "0.2" / >
              < /Appearance > < /Shape >
            < xsl : apply - templates select = ".//EXPLO" / >
          < /Group > < /Transform > < /Transform > < /xsl : for - each > < /Group >
        < /xsl : template >
      ...
    < /xsl : stylesheet >
```

6. Résultats

Nous avons implémenté une première version de notre formalisme et présentons dans cette section trois exemples permettant de démontrer la réutilisabilité des Génotypes et Phénotypes en présentant les résultats de différentes requêtes sur des ensembles de données distincts combinés avec différentes métaphores visuelles.

6.1. Carrousels de films

Cet exemple est basé sur l'exemple 1. Le Génotype est généré à partir des critères de recherche suivants :

- niveau d'exploration 1
 - critère hiérarchique : Genre
 - critère de métadonnées: nom
- niveau d'exploration 2
 - critère hiérarchique : Directeur
 - critère de métadonnées: couleur
- niveau d'exploration 3
 - critère hiérarchique : Titre du film
 - critère de métadonnées: url de l'affiche du film

Le texte passé en paramètre de chaque élément de la tour correspond à l'attribut *nom* du niveau d'exploration 1, tandis que les couleurs des sphères et des cadres correspondent à l'attribut *couleur* du niveau d'exploration 2. La texture des cadres correspond à l'attribut *poster* du dernier niveau d'exploration. Le résultat obtenu est présenté en figure 1.

6.2. Carrousels d'albums

Le Génotype utilisé dans cet exemple est généré à partir d'une base de donnée d'albums de musique, suivant les critères suivants :

- niveau d'exploration 1
 - critère hiérarchique : Genre
 - critère de métadonnées: nom
- niveau d'exploration 2
 - critère hiérarchique : Groupe
 - critère de métadonnées: couleur
- niveau d'exploration 3
 - critère hiérarchique : Titre de l'album
 - critère de métadonnées: url de la pochette

Le phénotype utilisé est le même que celui d'écrit dans l'exemple présenté en section 6.1. Le résultat obtenu est présenté en figure 6.

6.3. Cinémas

Dans cet exemple, nous utilisons la même base de données que dans l'exemple présenté en section 6.1. Les critères de hiérarchies sont les mêmes, et d'autres critères de métadonnées sont ajoutés (comme par exemple une couleur, une texture de sol et une texture de porte par genre).

Le Phénotype que nous avons défini est directement lié au type de données à visualiser: il s'agit d'un ensemble de scènes 3D représentant un univers de salles de cinéma. L'utilisateur peut naviguer dans ces scènes à l'aide d'un avatar. Le premier environnement correspond à une place sur laquelle des cinémas sont disposés de manière circulaire pour représenter chaque élément de niveau d'exploration 1 (genres). La couleur de la tour et du toit du cinéma sont définis à l'aide du Génotype. Un clic sur un bâtiment transporte

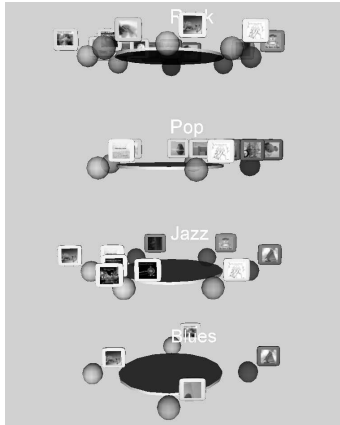
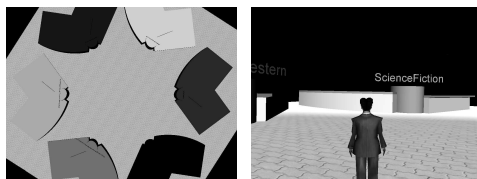
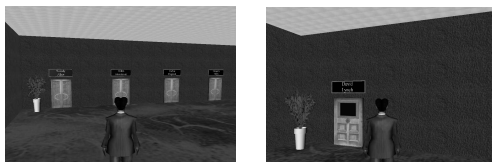


Figure 6: Phénotype 1: tour de carrousels représentant une base de données de musicale.

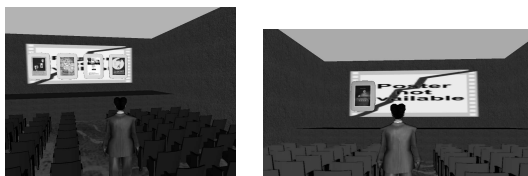
l'utilisateur dans le deuxième environnement représentant le hall du cinéma. Sur le mur du fond, les portes représentent les éléments de niveau d'exploration 2 (réalisateurs). Les textures du sol et des portes correspondent aux métadonnées du niveau d'exploration 1. Enfin, en cliquant sur une porte de son choix, l'utilisateur se retrouve dans une salle de cinéma à l'intérieur de laquelle les éléments (films) sont représentés sous forme de cadres. Le résultat obtenu est présenté en figure 7.



(a) Niveau d'exploration 1 : Genres.



(b) Niveau d'exploration 2 : Directeurs.



(c) Niveau d'exploration 3 : Films.

Figure 7: Phénotype 2 : Cinémas.

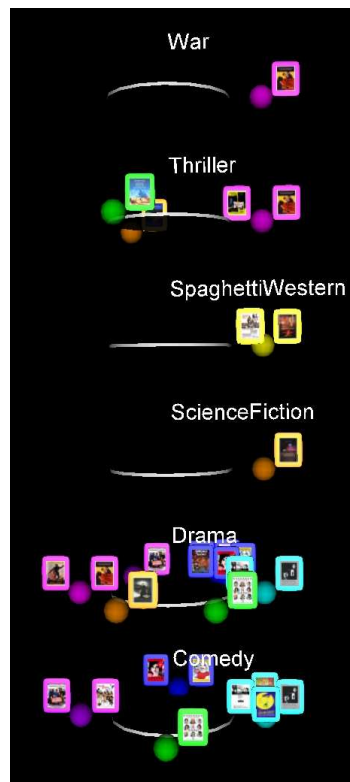
7. Conclusion et perspectives

Nous proposons un outil de création de métaphores de présentation de l'information basé sur deux composants : le module "Genotype computation" permettant de choisir la structuration exploratoire des données, et le module "Phenotype computation" permettant de définir l'aspect visuel de l'environnement 3D obtenu. Cette séparation entre les données et leur visualisation permet une forte réutilisabilité des Génotypes et Phénotypes afin de respectivement présenter les mêmes données avec différentes métaphores visuelles, et utiliser une unique métaphore visuelle pour présenter des données ou contenus différents. Notre approche étant basée sur les technologies Web, et en particulier XML, notre solution bénéficie de nombreux atouts (extensibilité, facilité d'échange et de transfert sur le réseau, ...) et est entièrement intégrée aux technologies et contenus Web. De plus, la généricité de notre solution lui permet d'être compatible avec les données issues de bases de données ou de résultats de recherche sémantique ou classique. Elle permet ainsi un passage plus facile au Web 3D tout en maîtrisant le style et la présentation de l'information.

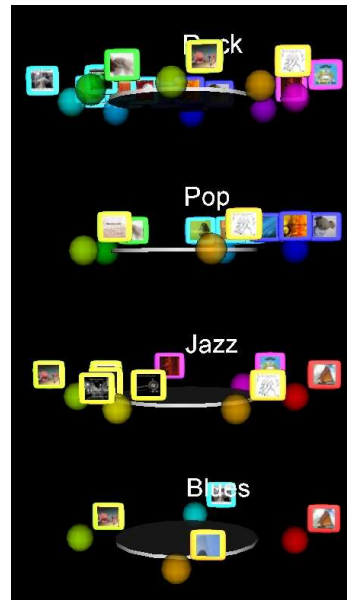
Par la suite, nous nous intéresserons au développement d'un module de définition de métaphores d'interaction et d'animation pouvant être combiné avec les modules présentés ici. Nous souhaitons également faciliter le choix du format de sortie, limité dans cet article à X3D. Enfin, nous souhaitons développer la partie usages pour la visualisation de réponses de moteurs de recherche sémantiques ou classiques, ainsi que la préservation d'éléments caractéristiques comme les couleurs, les images ou le style d'une marque.

References

- [BPSM*98] BRAY T., PAOLI J., SPERBERG-MCQUEEN C. M., MALER E., YERGEAU F.: Extensible markup language (xml) 1.0 (fourth edition), 1998.
- [Con04] CONSORTIUM W. W. W.: X3d specifications, 2004.
- [Kay07] KAY M.: Xsl transformations (xslt) version 2.0, January 2007.
- [SC07] SOMMARUGA L., CATENAZZI N.: Curriculum visualization in 3d. In *Web3D '07: Proceedings of the twelfth international conference on 3D web technology* (New York, NY, USA, 2007), ACM, pp. 177–180.
- [WCW06] WALCZAK K., CELLARY W., WHITE M.: Virtual museum exhibitions. *Computer* 39, 3 (2006), 93–95.
- [Web] WEB3D CONSORTIUM: Web 3d consortium.



(a) Exemple 1: films.



(b) Exemple 2: CDs.

Figure 8: Phénotype 1: tours de carrousels.